



PowerPlay Transformer Model Check List

Nature of Document: **Guideline**

Product(s): **PowerPlay Transformer**

Area of Interest: **Modeling**

Copyright and Trademarks

Licensed Materials - Property of IBM.

© Copyright IBM Corp. 2010

IBM, the IBM logo, and Cognos are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

While every attempt has been made to ensure that the information in this document is accurate and complete, some typographical errors or technical inaccuracies may exist. IBM does not accept responsibility for any kind of loss resulting from the use of information contained in this document. The information contained in this document is subject to change without notice.

This document is maintained by the Best Practices, Product and Technology team. You can send comments, suggestions, and additions to cscogpp@ca.ibm.com.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Table of Contents

Introduction.....4

 Purpose..... 4

 Applicability.....4

General Checks for the Transformer Model.....4

Data Sources.....5

 General Data Sources Checks:.....5

 Individual Data Sources Checks:..... 5

Dimensions.....7

Measures.....9

PowerCubes.....10

1 Introduction

1.1 Purpose

This guide gives a check list when quality reviewing a Transformer model. The check list is a suggested proven practice. By comparing your models against the items within this check list you will be able to go a fair way in stating that you have a good Transformer model that follows a number of proven practices.

These rules are not set in stone. Users may have valid reasons to deviate from these rules. Any deviations should be documented so that those following in your footsteps understand why standard proven practices have not been followed.

The guide does not state if a Transformer model is good or bad from a business perspective, nor if the resulting powercube will be efficient. Such judgement needs business knowledge. When performance tuning, knowing the business goal and designing a solution that better achieves these often gives a greater performance increase.

1.2 Applicability

The guide is applicable to all versions of IBM Cognos PowerPlay Transformer.

2 General Checks for the Transformer Model.

These checks relate to the entire Transformer model.

1. Use the **Tools > Check Model** menu item to easily identify any errors or warnings.

This tool is often overlooked yet it gives a good summary of any issues found with the Transformer model and you can easily identify whether any errors or warnings are acceptable.

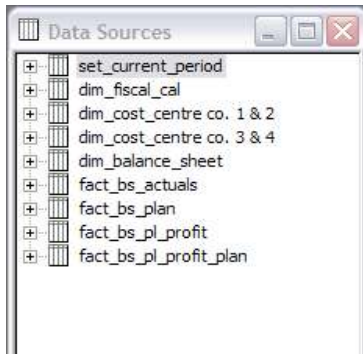
2. The model file is in **MDL format**.

The PY? format should only be used if model load time forms a significant part of the cube build times.

- An MDL version of the model ensures the model does not bloat or fragment over time when categories are deleted.
- Ensures forward compatibility that the model can be loaded into later versions of Transformer.
- N.B. For datasources that require passwords, MDL does not store these.

3 Data Sources.

This section deals with the data sources used within Transformer.



3.1 General Data Sources Checks:

1. Verify queries are split into dimension and fact queries.

An exception to this may be when a dimension is populated from a fact query. In this circumstance it may be foolish to read the fact table twice –once to get the dimension and again to get the facts. However if the query to get the distinct dimension list is of a low cost then two queries should be used.

2. Verify whether a naming convention is in place to easily identify fact and dimensional queries.
3. Verify dimensional queries appear before all fact queries.

In Cognos 8 Transformer the help documentation states that this is not necessary anymore however the way Transformer decides what is dimensional vs. fact has not been documented. This best practice should still be followed.

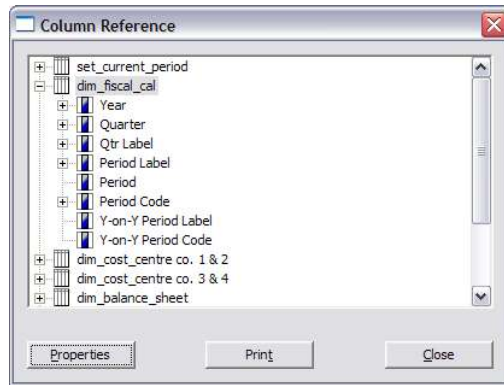
3.2 Individual Data Sources Checks:

1. Use the '**Show Scope**' tool.

You can use this tool on all Data Sources to ensure queries are only impacting expected dimensions. This tool can be found by right-clicking the data source and selecting 'Show Scope'.

2. Make sure no unreferenced columns are used.

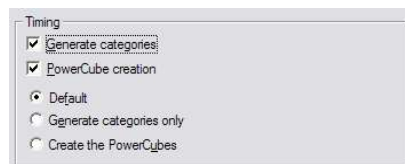
If a column is not used by model then we should not bring that data into Transformer. Right-click the data source and select the '**Show Reference**' tool to verify this.



If the data source is not a Framework Manager or a Cognos 8 report then remove unused columns from the underlying data source itself, not just from the Transformer query. For example if an IQD is used, it contains 10 columns but only 2 of those columns are used by Transformer then edit the IQD to remove the 8 unused columns.

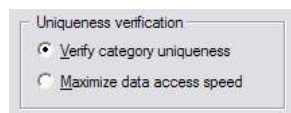
3. Check **Timing** attributes.

For dimensional queries only use **Generate Categories** or/and **PowerCube creation**
Generate Categories only.



For Fact queries, which do not populate dimensions, set the attributes to **Create the PowerCubes**.

4. If you are using a data warehouse where surrogates keys are guaranteed to be unique then consider **Maximise data access speed**.



5. Set the **current period** attribute only set on appropriate queries.

Consider having a query that does nothing but set the current period.

6. **Auto Summarize** property is checked /unchecked.

This setting depends upon the source data being read. If the source data is being used at the same granularity as the underlying table this should remain unchecked. If the source data is not consolidated then this should be checked.

Consider the effect this setting has upon the generated SQL of the data source .

– introducing summary functions.

- Ensure that the query has appropriate identifier and fact usage attributes set for this setting to be effective. These will need to be set in the source – either Framework Manager or the report. Again review the SQL to ensure appropriate grouping and summary functions are being applied.

7. Fact Data is consolidated.

If the auto summarize option is not available then ensure the fact query is consolidated. That is it only brings in one row of data for a unique key combination.

For example if an IQD is used you may have to edit the IQD to introduce appropriate Grouping and Summary functions.

8. Compare Native and Cognos SQL.

Ensure that the Cognos SQL is being turned into efficient Native SQL. For example use native sql functions where available.

4 Dimensions.

This section deals with the dimensions used within Transformer.

1. Category Codes are unique.

Use the Find Category option to search for category codes that contain a '~' character. Ensure this test is performed on a populated model.



The only category codes where this is acceptable is blank where blanks are suppressed. Having unpredictable category codes can impact MUNs and reports based upon them.

2. Calculated Categories / Special Categories do not use categories that themselves use Surrogate Keys / Unpredictable keys for their category codes.

There is no guarantee that such keys will be the same between Development, Test and Production systems. If these keys /codes change between environments or over time then Calculated Categories and Special Categories will become invalid.

3. Dimensions with alternate drill paths excluded from auto-partitioning.

Partitioning on the primary drill path could cause poor performance on the other hierarchy.

4. Flat dimensions excluded from auto-partitioning as they will not be useful for

partitioning.

5. Dimension levels that are 1:1.

If a category always drills to one and only one category then the two levels should be merged. A quick way to see if this is the case is to populate the model and check the category counts on the levels of the dimensions.

This is commonly seen when a data warehouse is used as a source and a level is created for the lowest level of the dimension and another level is created for the surrogate key level. The surrogate key level is often suppressed.

Either bring the fact in with the key of the dimension, or merge the two levels, making the SKKey the source and the appropriate field the name etc. Beware of the impact that this may have on previous points.

6. Lowest level of a dimension is unused.

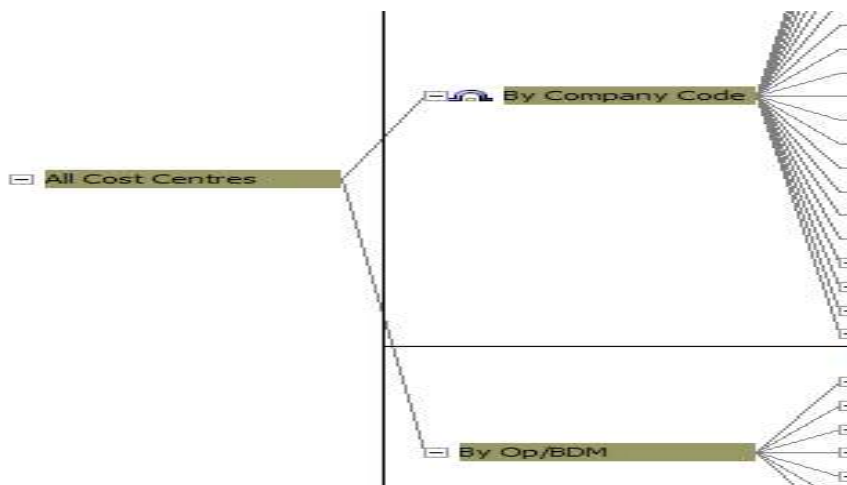
If the lowest level of a dimension is suppressed then is it really needed? This is a last resort.

If it has to be in the model and it is suppressed consider using the summarizing function on the level above. This will reduce cube sizes.

7. Scenario dimensions have a default category set.



8. Alternate Hierarchy root categories are given meaningful category labels and codes.



9. Dimensions which have the same or similar categories can be distinguished by the user.
Consider if a sales cube has Ordered Date and Shipped Date as dimensions. These two dimensions contain identical years, months and dates.
If a user places either of these dimensions on a report how would they know what month they are looking at?
10. All dimension levels are given meaningful names,
In the latest reporting studios level names are very visible to users.
11. If all categories within a level have the same category action applied.
If all categories within a level have a category action applies (for example exclude) then consider applying that action to the dimension level rather than the individual categories.
If the majority of categories have this applied then again consider applying to the dimension level and then un-actioning the other categories.
This statement is also applicable to custom views.
12. Category Uniqueness is only ticked on levels that are appropriate –not all levels.
13. Refresh options are only ticked when using a populated model.
If part of the build process is to generate all categories within the model then there is usually no need to refresh labels etc.

5 Measures.

This section deals with the measures used within Transformer.

14. Set an appropriate measure storage type.
If you are using a small set of the overall data when developing then beware of using a 32 bit measure – whilst this may be fine for development the total data volume may be too large for this data type.

64 bit measures require more storage so will result in larger powercube files. Only use a 64 bit measure if you are sure your data set requires it. When using 64 bit measures ensure appropriate scale and precision values are set.

This ensures the model can cope with large amounts of data. Ensure appropriate scale and precision values are set.
15. Appropriate Missing Value is set.
Consider the impact 'NA' may have on any calculated measures that may use this measure. Customers with prior versions of Transformer may prefer to set this to zero rather than the new default of NA.
16. Time state roll up of non-additive measures is set.
For example Closing Balance or Stock Level measures may be more appropriate to roll up using Current Period for actuals but Last Period for budgets.
17. Format is set.

18. Consider using an internal model name for the measure name and a user friendly Measure Label and Short Name. This give better maintainability of the model and any reports should the measure name need to change in the future.
19. Show Scope.
Use the 'Show Scope' tool on all measures to ensure scope is as expected.

6 PowerCubes.

This section deals with the PowerCubes used within Transformer.

1. PowerCube Partition status.
Check that the summary partition size is smaller than the desired partition size to ensure partitioning was successful.
2. Auto Partition.
On modern server architecture a desired partition size of 2.5m is a good starting point. Ensure the Estimated number of records reflects the environment the PowerCube will be eventually built.
Increase the Maximum number of passes to create smaller partitions should the build be unable to hit the desired partition size.
3. Remove the file path from the PowerCube filename and use the preference options to build in the appropriate location.
4. Enable Crosstab Caching on the cube Processing options.